A Systematic review of theory use in software engineering experiments

SUBMITTED BY:
GURPINDER KAUR
STUDENT ID-7960628

CONTENTS

Introduction

- Theories offer common conceptual frameworks that allow the structuring of knowledge in a concise and precise manner, thus facilitating the communication of ideas and knowledge.

- However, the usefulness of theories for software engineering is a subject of discussion. And the actual use of theory in empirical software engineering …?

- Empirical software engineering
- Software development must be investigated by empirical means to be understood, evaluated and deployed. So it is defined as a controlled experiment in SE in which teams conducts one or more SE tasks for the purpose of comparing treatments.

INTRODUCTION

- Motivation:

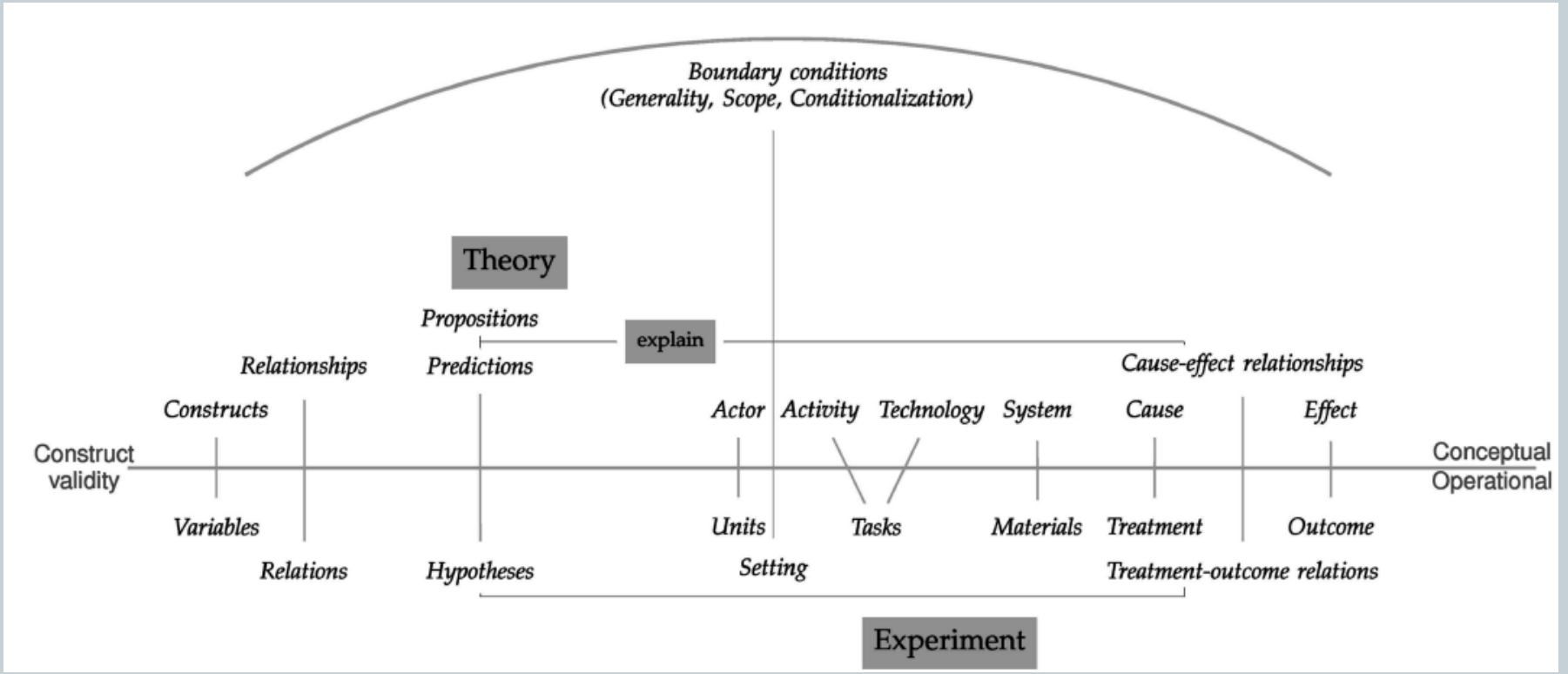
- An important undertaking in empirical software engineering is to determine what development technology to deploy and what developers to use in what situations. Or in other words we can say that the interest lies in comparing various technologies and skills and in determining their effects on software development.
- Whether and How Theory and empirical studies should interact if practical knowledge is to be acquired ??

TYPES OF THEORY

- Gregor's classification of theory:

1. Analysis- descriptions and conceptualizations of what it is ?
2. Explanation- explains why phenomenon occur but lack predictive power.
3. Prediction- predict without providing explanation eg. COCOMO model
4. Explanation and prediction- represents structures and why model is successful.
5. Design and action- describe how to do things and include design primitives.

Components of theories and experiments



Roles of theory

Following theory role categories have been described for the purpose of the review:

- Design
- Post hoc explanation
- Tested
- Modified
- Proposed
- Basis

RESEARCH METHOD

- Extraction of Experiments
- Extraction of Theory
- Extraction of theory attributes

Extraction of Experiments

- Assessed all the 103 articles describing experiments from the total of 5453 articles from 1993-2002.

Journal/Conference proceeding	*N*	%
Journal of Systems and Software (JSS)	24	23.3
Empirical Software Engineering (EMSE)	22	21.4
IEEE Transactions on Software Engineering (TSE)	17	16.5
International Conference on Software Engineering (ICSE)	12	11.7
IEEE International Symposium on Software Metrics (METRICS)	10	9.7
Information and Software Technology (IST)	8	7.8
IEEE Software	4	3.9
IEEE International Symposium on Empirical Software Engineering (ISESE)	3	2.9
Software Maintenance and Evolution (SME)	2	1.9
ACM Transactions on Software Engineering Methodology (TOSEM)	1	1.0
Software: Practice and Experience (SP&E)	-	-
IEEE Computer	-	-
Total	103	100

Extraction of theories

Theory inclusion criteria:

- Candidacy for theory

  The mention of the term theory or model or thereof together with at least one reference. OR

  The identification of constructs and relationships in a body of conceptual augmentation delineated by dia, words etc. AND

- Explanation of cause effect relationship

  The use in the roles of design, post hoc explanation, tested, modified, proposed or basis.

Extraction of theory attribute data

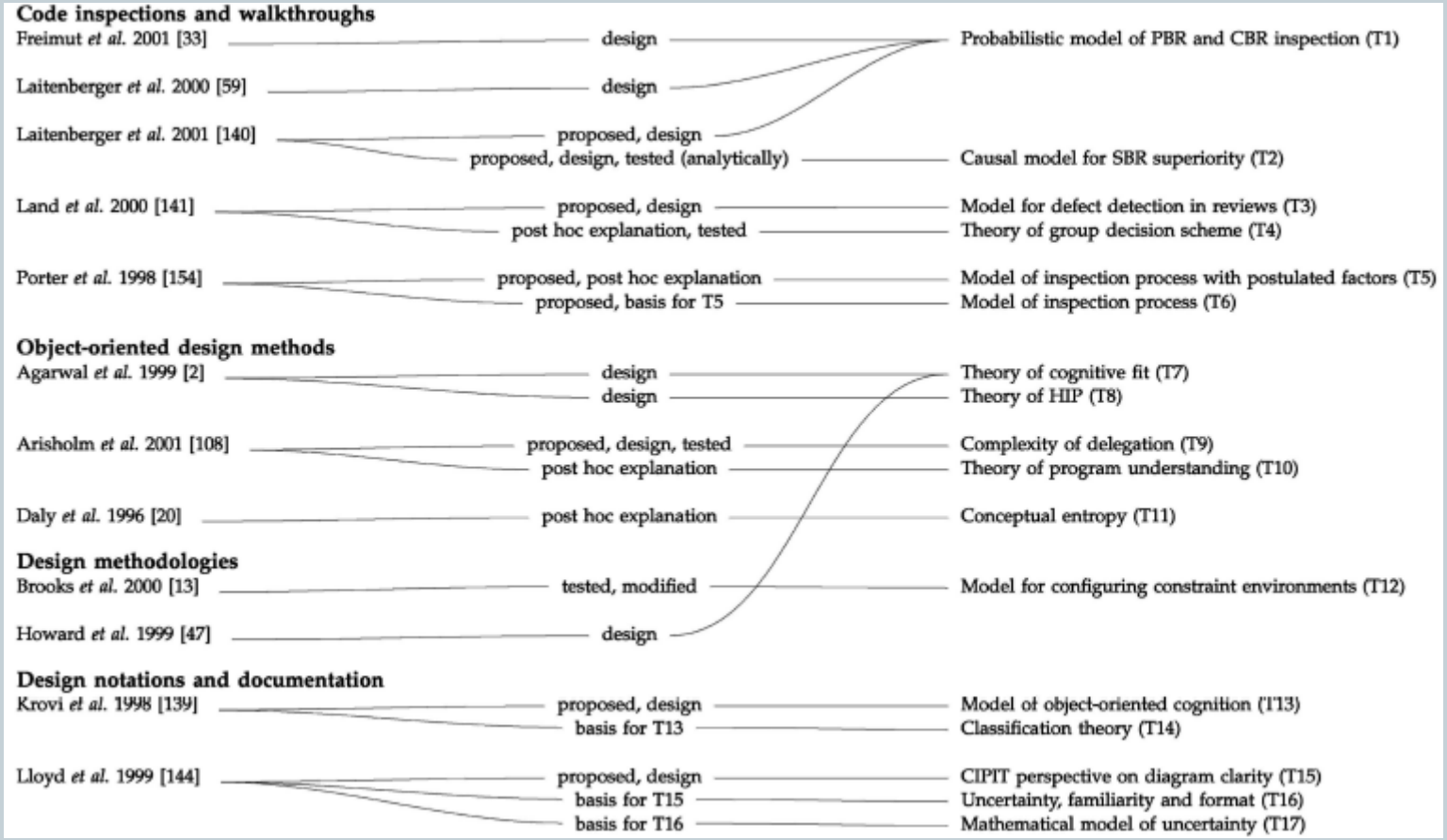
- Metadata - Name, references, topic, terminology like model, theory etc.

- Structural components

1. *Generic* – Constructs and relationships, boundary conditions, means of representations by word, table etc.
2. *Contingent on theory type* – casual explanation, prediction, prescription etc.

- Theory role – design, post hoc explanation, prediction etc.

FINDINGS


Code inspections and walkthroughs
Freimut et al. 2001 [33]
design
Probabilistic model of PBR and CBR inspection (T1)
Laitenberger et al. 2000 [59]
design
Laitenberger et al. 2001 [140]
proposed, design
proposed, design, tested (analytically)
Causal model for SBR superiority (T2)
Land et al. 2000 [141]
proposed, design
Model for defect detection in reviews (T3)
post hoc explanation, tested
Theory of group decision scheme (T4)
Porter et al. 1998 [154]
proposed, post hoc explanation
Model of inspection process with postulated factors (T5)
proposed, basis for T5
Model of inspection process (T6)
Object-oriented design methods
Agarwal et al. 1999 [2]
design
Theory of cognitive fit (T7)
design
Theory of HIP (T8)
Arisholm et al. 2001 [108]
proposed, design, tested
Complexity of delegation (T9)
post hoc explanation
Theory of program understanding (T10)
Daly et al. 1996 [20]
post hoc explanation
Conceptual entropy (T11)
Design methodologies
Brooks et al. 2000 [13]
tested, modified
Model for configuring constraint environments (T12)
Howard et al. 1999 [47]
design
Design notations and documentation
Krovi et al. 1998 [139]
proposed, design
Model of object-oriented cognition (T13)
basis for T13
Classification theory (T14)
Lloyd et al. 1999 [144]
proposed, design
CIPIT perspective on diagram clarity (T15)
basis for T15
Uncertainty, familiarity and format (T16)
basis for T16
Mathematical model of uncertainty (T17)

Structural components of theory

Component	*N*	%
Generic		
Means of Representation (other than name or references)	39	97.5
Constructs and Relationships	37	92.5
Boundary conditions	5	12.5
Contingent on Theory Type		
Causal explanations	32	80.0
Predictions	34	85.0
Prescriptive statements	0	0

Theory roles

- Theory as a motivation than incentive-- Most common use of theory is in the design role.
- Theory structuring – Several articles present theories that are present on other articles in a hierarchic manner.
- Theory generation

DISCUSSION

- Extent of theory use

  Not quantitatively but qualitatively can say yes.

  Basis of conceptual framework to which researchers may relate. Which is a prerequisite for building large cumulative bodies of knowledge.
- But overall cannot say theories are used for theoretical framework within which studies are conducted and interpreted. Rather used for supporting and motivating the study than the study being the result of theory.

- CONCLUSION:
- Theory driven investigations and theory building are rare in empirical software engineering.

  Constructive efforts- theory is proposed in only 12% of the 103 reviewed articles

Usefulness of theory

- Interesting
- If theory had not been used, some of the les obvious research questions investigated would have been harder to rationalize or even come by.
- Paths for new research in terms of underlying mechanisms which have not been investigated yet.

Obstacles of using theory

- Common conception that a massive body of empirical study must be accumulated prior to building theory.
- -But it can be derived mathematically or as an adaption of other theories etc

- Relating theory to that of empirical research.
- -Review can be useful in this regard since it summarizes how they can be used in experiments.

- Theory is abstract and therefore one must argue that it is of no use in software engineering.
- -Eg. understanding underlying mechanisms of software effort estimation which has direct impact on PM implications.

CONCLUSION

- Quarter of the articles surveyed involves theory.
- Mostly used to justify and motivate experimental research questions. However no theory driven research has been found.
- But theories are used and disciplines with strong theoretical traditions are being consulted when seeking explanation in empirical studies in SE.
- SO the author conquered with the view that the theories are the integral part of the empirical software engineering.

ANY QUESTIONS???